

Análise de Desempenho do CRAY Y-MP 2E/232

Jorge Luis Victória Barbosa ¹
Philippe Olivier Alexandre Navaux ²

Universidade Federal do Rio Grande do Sul -UFRGS
Instituto de Informática
Pós-Graduação em Ciência da Computação - CPGCC
Caixa Postal 15064 - CEP 91591-970
Porto Alegre, RS, Brasil

RESUMO

Este texto apresenta uma análise de desempenho do CRAY Y-MP 2E/232 instalado atualmente no Centro Nacional de Supercomputação (CESUP) administrado pela Universidade Federal do Rio Grande do Sul (UFRGS). Basicamente, são descritos e analisados os resultados obtidos em diversos testes realizados durante a execução de um programa altamente vetorizável. O texto descreve os programas utilizados nos testes e discute questões relacionadas com vetorização. Durante essa discussão são abordados vários aspectos que influenciam o desempenho do CRAY dentre os quais destacam-se: ordem dos índices nos laços para tratamento de matrizes, preenchimento incompleto dos registradores vetoriais, forma de alocação dos vetores em memória e tamanho dos vetores processados.

ABSTRACT

This text presents a performance analysis of the CRAY Y-MP 2E/232 which is installed at the "Centro Nacional de Supercomputação" (CESUP) managed by the "Universidade Federal do Rio Grande do Sul" (UFRGS). Basically, the results obtained in several tests performed during the execution of a highly vectorized program are described and analysed. The text describes the programs used in these tests and examines several questions related to the vectorization. During this discussion several aspects that affect the performance of the CRAY are approached. Some of these are: the order of the indices in matrix treatment, incomplete fulfillment of vector registers, vector allocation in memory and size of the vectors processed.

¹Professor na Universidade Católica de Pelotas (UCPel/RS), Tecnólogo em Processamento de Dados (UCPel/RS, 1989), Engenheiro Eletricista (UCPel/RS, 1990), Especialista em Engenharia de Software (UCPel/RS, 1992), Mestrando do CPGCC/UFRGS. Áreas de Interesse: Processamento Paralelo, Paralelismo na Programação em Lógica e Avaliação de Desempenho. E-mail: barbosa@atlas.ucpel.tche.br

²Professor na Universidade Federal do Rio Grande do Sul (UFRGS/RS), Doutor em Engenharia de Informática (Instituto Nacional Politécnico de Grenoble, França, 1979). Áreas de Interesse: Arquitetura de Computadores, Processamento Paralelo e Avaliação de Desempenho. E-mail: navaux@inf.ufrgs.br

1 INTRODUÇÃO

Devido a seu alto poder computacional e seu direcionamento para processamento numérico pesado, os supercomputadores vêm sendo utilizados em laboratórios de pesquisa, universidades e empresas, para resolver problemas complexos que envolvam grandes estruturas de dados e grande quantidade de cálculos. Atualmente, existem diversas aplicações que utilizam o potencial desses equipamentos. Dentre essas aplicações destacam-se: previsão meteorológica, pesquisas atmosféricas e oceânicas, mecânica quântica, dinâmica molecular, exploração e prospecção de petróleo e modelagem econômica.

O desenvolvimento de supercomputadores envolve diversas técnicas avançadas que permitem a obtenção de alto desempenho no processamento de dados. Atualmente, o **processamento vetorial** ([NAV90]) destaca-se como uma dessas técnicas, sendo amplamente utilizado no projeto de poderosos sistemas computacionais. A utilização de técnicas avançadas para projeto de computadores de alto desempenho implica na criação de novas arquiteturas de computadores (*hardware*) e novos programas de sistema (*software*). No caso do processamento vetorial, destaca-se o surgimento dos processadores vetoriais e dos compiladores vetorizadores ([ZIM91]).

Neste universo de técnicas inovadoras, o conhecimento das características introduzidas pelas novas arquiteturas e novos programas de sistema é essencial para obtenção de altos níveis de desempenho. Por exemplo, conforme discutido em [BAR92a], no âmbito do processamento vetorial existem diversos **métodos de vetorização** que devem ser considerados pelo usuário durante o desenvolvimento de programas. Além disso, o conhecimento da arquitetura dos computadores vetoriais, mostra-se em determinadas situações essencial para agilizar o processamento.

No mês de junho do ano de 1992, o Centro Nacional de Supercomputação (CESUP), administrado pela Universidade Federal do Rio Grande do Sul (UFRGS), iniciou a operação de um supercomputador modelo CRAY Y-MP 2E/232. Atualmente, esse equipamento está sendo utilizado no desenvolvimento de aproximadamente 250 projetos. Uma das principais características dos sistemas computacionais CRAY consiste no uso de processadores vetoriais.

Este texto apresenta uma análise de desempenho do CRAY instalado atualmente no CESUP, realizando uma série de considerações quanto a resultados obtidos na execução de um programa de teste altamente vetorizável (produto matricial). Durante a apresentação desses resultados são discutidos diversas características que podem ser consideradas pelos usuários do CRAY para aperfeiçoamento de seus programas. O principal objetivo desse trabalho é a difusão de informações que auxiliem os usuários dos supercomputadores CRAY a obterem o máximo desempenho em suas aplicações. O texto está organizado em 5 seções. Na seção 2 é apresentada uma descrição do produto matricial e do algoritmo utilizado na análise de desempenho do CRAY, realizando-se considerações quanto a sua lógica e vetorização. Na seção 3 é realizada uma descrição do CRAY Y-MP instalado no CESUP, descrevendo-se resumidamente sua arquitetura e alguns softwares disponíveis. Na seção 4 são descritos os testes realizados na análise de desempenho, apresentando-se os resultados obtidos e descrevendo-se os softwares utilizados. Finalizando, a seção 5 apresenta as conclusões deste trabalho.

2 PRODUTO MATRICIAL

As informações apresentadas nesta seção foram obtidas em [BAR92a] e [BAR92b]. Basicamente, descreve-se o produto matricial enfatizando-se algumas características relevantes para compreensão do restante do texto.

2.1 DESCRIÇÃO

A multiplicação de matrizes constitui um dos principais estágios de computação para solucionar os mais diversos problemas. No processamento seqüencial, os algoritmos para resolver este tipo

específico de problema requerem uma grande quantidade de memória e um consumo de tempo relativamente alto.

Por outro lado, a multiplicação de matrizes é um típico problema para ser implementado em computadores vetoriais. O produto de uma matriz A ($m \times n$) por uma matriz B ($n \times p$) é uma matriz C ($m \times p$) cujos elementos são definidos por:

$$c(i,j) = \sum_{k=1}^n a(i,k) \cdot b(k,j) \quad \text{onde } i = 1, 2, \dots, m \\ j = 1, 2, \dots, p$$

Por exemplo, a multiplicação de uma matriz A (2×2) por uma matriz B (2×2), resulta numa matriz C (2×2):

$$A \cdot B = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \cdot \begin{vmatrix} e & f \\ g & h \end{vmatrix} = \begin{vmatrix} a \cdot e + b \cdot g & a \cdot f + b \cdot h \\ c \cdot e + d \cdot g & c \cdot f + d \cdot h \end{vmatrix} = C$$

Nota-se que o elemento da primeira linha e primeira coluna da matriz C é obtido pela multiplicação de cada elemento da primeira linha da matriz A pelo correspondente elemento na primeira coluna da matriz B e adicionando-se os resultados. Este efeito é conhecido como "produto linha vezes coluna", ou seja, quando calcula-se o elemento "a.e + b.g", na verdade, está se multiplicando a primeira linha de A pela primeira coluna de B . Similarmente, o elemento "a.f + b.h" na primeira linha e segunda coluna de C é a primeira linha de A multiplicada pela segunda coluna de B ; e assim sucessivamente. Desta forma todos os elementos de C podem ser calculados pela utilização da seguinte regra:

O elemento na linha i e coluna j do produto matricial $A \cdot B$ é igual à linha i da matriz A multiplicada pela coluna J da matriz B .

Considere o exemplo a seguir.

$$A = \begin{vmatrix} 4 & 2 \\ -3 & 1 \end{vmatrix} \quad B = \begin{vmatrix} 1 & 5 & 3 \\ 2 & 7 & -4 \end{vmatrix}$$

$$C = A \cdot B = \begin{vmatrix} 4 & 2 \\ -3 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 5 & 3 \\ 2 & 7 & -4 \end{vmatrix}$$

$$= \begin{vmatrix} 4 \cdot 1 + 2 \cdot 2 & 4 \cdot 5 + 2 \cdot 7 & 4 \cdot 3 + 2 \cdot (-4) \\ (-3) \cdot 1 + 1 \cdot 2 & (-3) \cdot 5 + 1 \cdot 7 & (-3) \cdot 3 + 1 \cdot (-4) \end{vmatrix} =$$

$$= \begin{vmatrix} 8 & 34 & 4 \\ -1 & -8 & -13 \end{vmatrix}$$

Torna-se importante ressaltar, que o produto entre duas matrizes A e B somente é possível se o número de colunas da matriz A é igual ao número de linhas da matriz B .

Conforme apresentado na subseção 2.2, os elementos $C(i,j)$ podem ser calculados utilizando-se três laços computacionais: um interno, um intermediário e um externo. O laço interno deverá conter a expressão:

$$C(i,j) = C(i,j) + a(i,k) \cdot b(k,j)$$

Vários algoritmos podem ser utilizados para implementar o produto entre duas matrizes. A próxima subseção discute o algoritmo utilizado para realizar os testes apresentados neste trabalho.

2.2 ALGORITMO UTILIZADO NA ANÁLISE DE DESEMPENHO

Segundo [QUI87], o algoritmo apresentado na figura 2.1, proporciona uma das formas mais eficientes para a realização do produto em computadores vetoriais, de uma matriz *A* com *l* linhas e *m* colunas, por uma matriz *B* com *m* linhas e *n* colunas, resultando uma matriz *C* com *l* linhas e *n* colunas.

```

for i ← 0 to l-1 do
  for j ← 0 to n-1 do
    C(i,j) ← 0
  endfor
endfor
for j ← 0 to n-1 do
  for k ← 0 to m-1 do
    for i ← 0 to l-1 do
      C(i,j) ← C(i,j) + A(i,k) x B(k,j)
    endfor
  endfor
endfor

```

Figura 2.1 - Algoritmo para produto matricial em computadores vetoriais

Nota-se no algoritmo a utilização de dois laços para colocar zero em todos os elementos da matriz resultante, pois estes serão utilizados como acumuladores dos resultados durante a multiplicação. A independência destes laços em relação aos demais, permite a vetorização do laço mais interno desta operação.

Analisando-se os três laços utilizados para realizar o produto, nota-se a colocação do laço de índice *i* como mais interno, possibilitando a utilização de uma operação vetorial no formato $V \leftarrow V + V \times S$, onde *V* é um operando vetorial e *S* um operando escalar. O mesmo resultado poderia ser alcançado com a colocação do laço de índice *j* como o mais interno, mas não com a colocação do laço de índice *k*.

Cabe ressaltar que a ordem dos laços numa multiplicação de matrizes não altera o resultado final, possibilitando uma grande flexibilidade na organização dos laços para facilitar a vetorização.

Deve-se ressaltar ainda, que a operação vetorial $V \leftarrow V + V \times S$, permite a utilização de encadeamento entre as operações de soma e multiplicação, o que pode aumentar de forma considerável o desempenho do algoritmo.

3 DESCRIÇÃO DO CRAY INSTALADO NO CESUP

O CRAY Y-MP 2E/232 atualmente instalado no Centro Nacional de Supercomputação administrado pela UFRGS possui a seguinte configuração:

- Dois processadores de 330 Mflops
- Capacidade da Memória de 32 Mwords (256 Mbytes / 4 seções)
- Capacidade de armazenamento em disco de 16 Gigabytes
- Uma estação SUN 4/370 como Front-End
- Uma estação SUN 4/370 como Unidade de Controle e Manutenção

Dentre os softwares atualmente instalados no CRAY destacam-se:

- Sistema Operacional UNICOS versão 6.1.5a+
- Compiladores Fortran, C e Pascal
- Linguagem Assembly
- Diversos utilitários (UNICHEM, MPGS, ANSYS, SPICE, etc)
- Utilitários para análise de desempenho de programas (HPM, etc)

Levando-se em consideração que os programas desenvolvidos para computadores vetoriais são altamente dependentes da arquitetura do equipamento utilizado, apresenta-se a seguir algumas das características relevantes de uma CPU do CRAY Y-MP ([BAR93]):

- 14 unidades funcionais pipeline (5 para operações vetoriais, 4 para operações escalares, 2 para operações de endereços e 3 para operações com ponto-flutuante)
- 3 registradores primários (registradores escalares, registradores de endereços e registradores vetoriais)
- 2 registradores secundários (registradores intermediários de endereços e registradores intermediários escalares)
- Os registradores vetoriais podem armazenar 64 elementos de 64 bits (uma word)

4 DESCRIÇÃO DOS TESTES E RESULTADOS OBTIDOS

4.1 DESCRIÇÃO DOS TESTES

Os testes realizados para avaliar o desempenho do CRAY Y-MP 2E/232, consistiram na execução de programas implementados na linguagem C, seguindo-se o algoritmo apresentado na figura 2.1 e alterando-se algumas das suas características, tais como as dimensões das matrizes e a ordem dos laços utilizados para realizar o produto. O item 4.2 apresenta a estrutura do programa e descreve os softwares utilizados no teste de desempenho.

Além das alterações nas características do algoritmo, cada um dos testes foi realizado vetorizando-se e não vetorizando-se o programa, o que possibilitou uma avaliação das diferenças entre o processamento escalar e vetorial no CRAY Y-MP.

Para cada um dos testes obteve-se o tempo de execução do programa e o número de operações de ponto-flutuante por segundo (MFlops), os quais foram utilizados como parâmetros para a realização da análise de desempenho.

O primeiro teste consistiu na multiplicação de duas matrizes quadradas de 700 linhas e 700 colunas, alterando-se a ordem dos laços da multiplicação, ou seja, foram realizadas seis produtos matriciais, testando-se todas as possíveis combinações de índices dos laços. O objetivo desta análise foi avaliar a influência da ordem de busca dos elementos na memória sobre o desempenho do produto matricial. Os resultados obtidos são apresentados na subseção 4.3.

O segundo teste consistiu na multiplicação de matrizes quadradas com dimensões variáveis, objetivando-se avaliar o desempenho do CRAY Y-MP neste tipo de operação. Os resultados são discutidos na subseção 4.4.

O terceiro teste consistiu na multiplicação de uma matriz com 700 linhas e 700 colunas por outra matriz com 700 linhas e um número variável de colunas. O objetivo deste teste foi avaliar a influência do preenchimento dos registradores vetoriais no desempenho do produto matricial. Os resultados são apresentados na subseção 4.5.

No quarto e último teste foi realizada uma comparação do desempenho do CRAY com vários tipos de estações SUN, quando submetidos a multiplicação de duas matrizes quadradas com 500 linhas e 500 colunas. A subseção 4.6 apresenta os resultados obtidos nesta análise.

4.2 SOFTWARES UTILIZADOS NOS TESTES

A estrutura básica do programa utilizado para realização dos testes descritos nas subseções 4.3, 4.4, 4.5 e 4.6 é apresentado na figura 4.1. Os testes descritos nas próximas subseções alteram algumas características desse programa, o qual é especificamente para multiplicação de matrizes quadradas de 700 linhas e 700 colunas. No entanto, sua estrutura principal permanecerá inalterada, mudando-se apenas a ordem dos índices dos três laços do produto matricial (primeiro teste) e as dimensões das matrizes (segundo, terceiro e quarto testes).

Utilizou-se o compilador C disponível no sistema operacional UNICOS do CRAY ([CRA91a]), o qual possibilita a vetorização automática dos programas. A compilação de um programa fonte denominado *mul_mat.c* utilizando vetorização automática é realizada com a seguinte linha de comando:

```
cc mul_mat.c -o mul_mat.v
```

Neste caso, o arquivo executável vetorizado recebe o nome *mul_mat.v*.

A compilação de um programa fonte com a opção de vetorização desligada é realizada com a seguinte linha de comando:

```
cc -h vector0 mul_mat.c -o mul_mat.nv
```

Neste caso, o programa executável não vetorizado recebe o nome *mul_mat.nv*.

```
main()
{
  float mat1[700][700], mat_2[700][700], mat_r[700][700];
  int i,j,k;

  for(i = 0; i < 700; i++)
    for(j = 0; j < 700; j++)
      mat_r[i][j] = 0;

  for(i = 0; i < 700; i++)
    for(k = 0; k < 700; k++)
      for(j = 0; j < 700; j++)
        c[i][j] = c[i][j] + a[i][k] * b[k][j];
}
```

Figura 4.1 - Programa utilizado nos testes do CRAY

A análise de desempenho dos programas executáveis gerados com o compilador C foi realizada com o utilitário HPM (*Hardware Performance Monitor*)([CRA91b]), o qual fornece uma série de informações relacionadas com a execução de um programa. Dentre as informações apresentadas pelo HPM podem ser encontrados o tempo de execução e o número de operações de ponto-flutuante por segundo (Mflops). A seguinte linha de comando demonstra como utilizar o HPM para monitorar a execução do programa *mul_mat.v* gerado pelo compilador C.

```
hpm mul_mat.v
```

4.3 MUDANÇA NA ORDEM DOS ÍNDICES

Inicialmente, deve-se lembrar que dentre um conjunto de laços aninhados, apenas o mais interno pode ser vetorizado. Além disso, as linguagens possuem características próprias para o armazenamento de matrizes em memória, o que normalmente varia dependendo da linguagem.

A linguagem FORTRAN armazena os elementos de uma matriz, seqüencialmente, coluna após coluna. Por exemplo, uma matriz A com 4 linhas e 4 colunas possuiria a seguinte ordem seqüencial no armazenamento dos elementos:

```
A(1,1)
A(2,1)
A(3,1)
A(4,1)
A(1,2)
A(2,2)
A(3,2)
A(4,2)
A(1,3)
A(2,3)
A(3,3)
A(4,3)
A(1,4)
A(2,4)
A(3,4)
A(4,4)
```

Por outro lado, a linguagem C armazena os elementos de uma matriz, seqüencialmente, linha após linha, por exemplo, uma matriz A com 4 linhas e 4 colunas possuiria a seguinte ordem seqüencial no armazenamento de seus elementos na memória.

```
A(1,1)
A(1,2)
A(1,3)
A(1,4)
A(2,1)
A(2,2)
A(2,3)
A(2,4)
A(3,1)
A(3,2)
A(3,3)
A(3,4)
A(4,1)
A(4,2)
A(4,3)
A(4,4)
```

Torna-se importante ressaltar ainda, que o CRAY Y-MP 2E/232 possui 4 seções de memória, 2 portas para leitura e uma porta para escrita em memória por cada CPU, podendo realizar estas operações simultaneamente. Uma CPU não pode realizar dois acessos simultâneos na mesma seção de memória, o que ocasiona conflitos e atrasos na execução de programas. Além disso, os elementos de uma matriz serão distribuídos seqüencialmente pelas 4 seções, ou seja, na linguagem C os elementos da matriz utilizada no exemplo possuiriam a seguinte distribuição:

<u>SEÇÃO 0</u>	<u>SEÇÃO 1</u>	<u>SEÇÃO 2</u>	<u>SEÇÃO 3</u>
A(1,1)	A(1,2)	A(1,3)	A(1,4)
A(2,1)	A(2,2)	A(2,3)	A(2,4)
A(3,1)	A(3,2)	A(3,3)	A(3,4)
A(4,1)	A(4,2)	A(4,2)	A(4,4)

Portanto, a mudança na ordem dos índices dos laços do produto matricial, apesar de não influenciar nos resultados finais, pode influenciar diretamente no desempenho do programa, pois reflete nos conflitos de acesso a memória.

Sabe-se ainda, que a leitura seqüencial dos elementos de uma matriz armazenada na memória, diminui substancialmente os conflitos. Portanto, durante o desenvolvimento de um programa a ser vetorizado, deve-se colocar no laço mais interno, o índice que possibilitará a realização de uma leitura seqüencial na memória dos elementos das matrizes. Os resultados obtidos com as mudanças na ordem dos índices no produto de duas matrizes quadradas com 700 linhas e 700 colunas é apresentado na tabela 4.1.

TABELA 4.1 - PRODUTO COM MUDANÇA NA ORDEM DOS ÍNDICES

ORDEM DO ÍNDICES	TEMPO NÃO VETORIZADO	MFLOPS NÃO VETORIZADO	TEMPO VETORIZADO	MFLOPS VETORIZADO
i/k/j	43.05	15.94	3.09	222.30
k/i/j	42.95	15.97	3.09	222.29
i/j/k	29.51	23.25	4.22	170.80
j/i/k	29.61	23.17	4.23	170.69
j/k/i	41.67	16.46	4.93	139.16
k/j/i	42.40	16.18	4.94	138.99

A primeira coluna indica a ordem dos índices dos laços do produto matricial, sendo o índice da esquerda o externo e o índice da direita o interno. A tabela foi organizada em ordem decrescente de *Mflops* alcançados com vetorização.

Avaliando-se os resultados, nota-se que o maior desempenho foi alcançado quando o índice *j* é colocado no laço mais interno (laço vetorizado) e que o menor desempenho foi alcançado quando coloca-se o índice *i* no laço mais interno.

Levando-se em consideração que os testes foram realizados com a utilização da linguagem C e portanto os elementos das matrizes foram armazenados seqüencialmente por linhas, a colocação do índice *j* no laço mais interno diminuiu o número de conflitos de acesso a memória, pois os elementos serão indexados por coluna, possibilitando uma leitura seqüencial na memória.

Avaliando-se os resultados, pode-se verificar que a colocação do índice *i* no laço mais interno obteve o pior desempenho, ocasionado pelos conflitos de acesso a memória gerados pela indexação por linha.

Pode-se verificar ainda nos resultados do teste, a diferença de desempenho obtido com a utilização da vetorização e sem sua utilização. Analisando-se a primeira linha da tabela pode-se verificar uma variação de 15.94 *Mflops* para 222.30 *Mflops*, o que equivale a um ganho de desempenho de aproximadamente 14 vezes.

Nos testes apresentados nos itens 4.4, 4.5 e 4.6, os índices foram ordenados conforme a primeira linha da tabela 4.1, ou seja *i/k/j*.

4.4 PRODUTO DE MATRIZES QUADRADAS

Neste teste foi realizado o produto de matrizes quadradas, variando-se o número de colunas e linhas, crescentemente de 100 até 700. Os resultados obtidos são apresentados na tabela 4.2.

Analisando-se a quinta coluna da tabela 4.2, verifica-se que o aumento do número de linhas e colunas das matrizes quadradas a serem multiplicadas, ocasionou um aumento do desempenho obtido pelo programa. No entanto, nota-se que o aumento não é linear, por exemplo, a diferença de desempenho entre o produto das matrizes com 200 linhas e colunas e o produto das matrizes de 100 linhas e colunas é de 26.84 Mflops. No entanto, a diferença de desempenho entre o produto das matrizes com 400 linhas e colunas e o produto das matrizes de 500 linhas e colunas é de 2.84 Mflops. Avaliando-se essa situação, pode-se concluir que dois são os principais fatores que influenciaram no desempenho alcançado nos produtos das matrizes quadradas.

TABELA 4.2 - PRODUTO DE MATRIZES QUADRADAS

NÚMERO DE COLUNAS E LINHAS	TEMPO NÃO VETORIZADO	MFLOPS NÃO VETORIZADO	TEMPO VETORIZADO	MFLOPS VETORIZADO
100	0.13	15.02	0.01	162.41
200	1.05	15.22	0.08	189.25
300	3.53	15.28	0.26	204.18
400	8.36	15.31	0.62	207.02
500	15.47	16.16	1.16	215.21
600	26.95	16.03	1.99	216.72
700	43.05	15.94	3.09	222.30

O primeiro é o número de iterações do laço mais interno, ou seja, quanto maior o número de colunas da segunda matriz, maior será o desempenho obtido no produto, pois a instrução vetorial processará vetores maiores. O segundo fator a ser considerado é o preenchimento do registrador vetorial, levando-se em consideração o número de elementos da operação vetorial (laço mais interno). Os registradores vetoriais do CRAY podem armazenar 64 elementos, portanto quanto maior o preenchimento obtido pela última carga da operação vetorial, maior o desempenho do programa. Por exemplo, na multiplicação de matrizes quadradas de 700 linhas e colunas, a operação vetorial obtida no laço mais interno preencherá completamente o registrador vetorial do CRAY 10 vezes. No entanto, restaram 60 elementos a serem processados, ocasionando o preenchimento parcial do registrador vetorial, com 4 elementos a menos de sua capacidade.

Por outro lado, o produto de matrizes com 400 linhas e colunas, ocasionará o preenchimento completo do registrador vetorial 6 vezes, sendo necessária mais uma carga do registrador com apenas 16 elementos, o que ocasiona perda de desempenho. Os dois fatores combinados ocasionam a falta de linearidade no aumento do desempenho apresentado na tabela 4.2.

4.5 PREENCHIMENTO DOS REGISTRADORES VETORIAIS

O terceiro teste avalia a influência do preenchimento dos registradores vetoriais no desempenho do produto matricial. Foi realizado o produto de uma matriz quadrada com 700 linhas e 700 colunas por outra matriz com 700 linhas e um número variável de colunas. O número de colunas da segunda matriz reflete diretamente no preenchimento do registrador vetorial, pois o índice do laço mais interno (vetorizado) é utilizado para indexar na memória as colunas da segunda matriz. A tabela 4.3 apresenta os resultados obtidos neste teste.

A primeira coluna da tabela 4.3 apresenta o número de colunas da segunda matriz do produto matricial. Foram utilizados valores entre 1 e 700, onde na faixa entre 1 e 200 utilizou-se valores crescentes com uma variação de 20 e entre 200 e 700 utilizou-se uma variação de 100. Nos valores entre 1 e 200 foram detalhados os pontos onde ocorre o preenchimento completo do registrador vetorial, ou seja, valores múltiplos de 64 (64, 128 e 192), onde analisou-se ainda, o valor imediatamente posterior (65, 129 e 193), o qual ocasiona uma carga do registrador vetorial para apenas um elemento.

Analisando-se os resultados apresentados na tabela 4.3, pode-se verificar que na medida que aumenta-se o número de colunas da segunda matriz, aumenta o desempenho do produto matricial. Nota-se que o desempenho aumenta continuamente até ocorrer um valor múltiplo de 64, onde ocorre uma queda de desempenho no valor imediatamente posterior. Por exemplo, quando o número de colunas da segunda matriz alcançou 64, obteve-se 150.36 Mflops, ocorrendo uma queda de desempenho (135.09 Mflops) no valor imediatamente posterior (65). Esta queda é ocasionada pela necessidade de mais uma carga do registrador vetorial para apenas um elemento. Após a queda ocasionada por este evento, o desempenho cresce novamente, obtendo-se inclusive, valores acima dos índices já alcançados. A queda de desempenho ocorrerá sempre no valor imediatamente consecutivo aos múltiplos de 64.

TABELA 4.3 - PREENCHIMENTO DOS REGISTRADORES VETORIAIS

NÚMERO DE COLUNAS DA SEGUNDA MATRIZ	TEMPO NÃO VETORIZADO	MFLOPS NÃO VETORIZADO	TEMPO VETORIZADO	MFLOPS VETORIZADO
1	0.30	3.32	0.30	3.32
20	1.70	11.54	0.27	73.86
40	3.28	11.97	0.33	117.87
60	4.85	12.12	0.41	144.80
64	5.58	11.24	0.42	150.36
65	4.11	15.52	0.47	135.09
80	6.53	12.00	0.56	140.05
100	6.50	15.09	0.60	164.46
120	8.10	14.52	0.65	181.46
128	9.39	13.35	0.66	190.29
129	7.92	15.97	0.74	171.42
140	9.70	14.15	0.78	176.16
160	11.30	13.88	0.84	186.42
180	11.26	15.67	0.90	195.34
192	13.21	14.25	0.93	203.14
193	11.73	16.12	1.02	185.47
200	12.86	15.24	1.03	191.19
300	19.20	15.31	1.44	204.29
400	25.55	15.34	1.88	209.07
500	30.28	16.18	2.30	213.17
600	36.62	16.06	2.70	217.90
700	43.05	15.94	3.09	222.30

A figura 4.2 apresenta um gráfico onde pode-se verificar a curva que descreve o desempenho do produto matricial na faixa de 0 até 200. Analisando-se o gráfico, constata-se as quedas de desempenho ocasionadas pelo preenchimento parcial do registrador vetorial.

Analisando-se a tabela 4.3, pode-se verificar ainda, que o desempenho global do produto matricial cresce continuamente, apesar das quedas ocasionadas pelo preenchimento parcial dos registradores vetoriais. Inicialmente, o desempenho cresce bruscamente, tendendo a estabilizar na medida que aumenta-se o número de colunas da segunda matriz.

A figura 4.3 apresenta um gráfico onde constata-se o crescimento contínuo do desempenho do produto matricial entre os valores de 0 à 700. Neste gráfico, destaca-se o rápido crescimento inicial do desempenho até aproximadamente 100 iterações e a tendência a estabilização próximo aos 222 Mflops. Segundo [HWA85], quanto maior o vetor submetido ao *pipeline* (número de colunas da segunda matriz), menos representativo será o *overhead* na realização da operação.

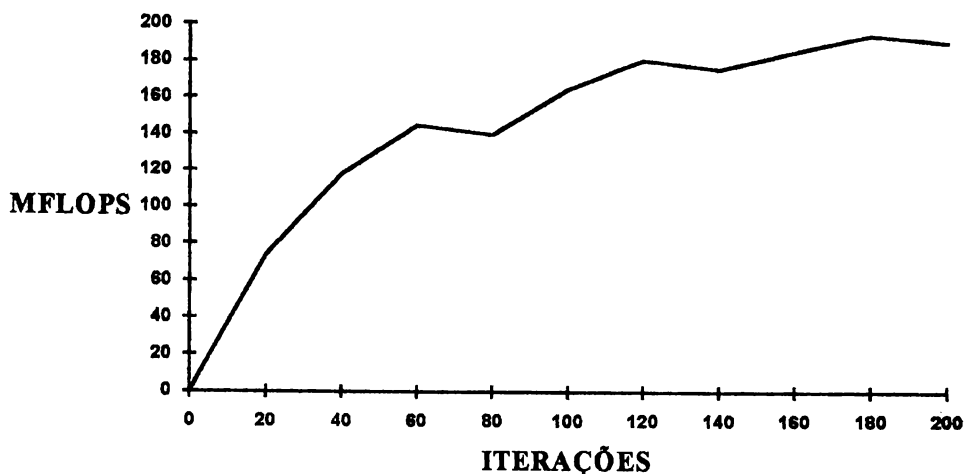


Figura 4.2 - Gráfico Mflops X Iterações (0 à 200)

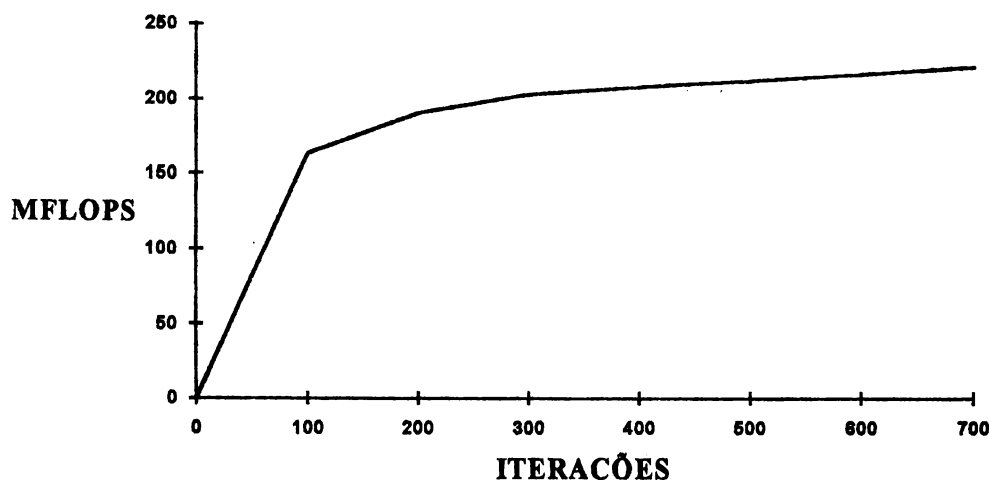


Figura 4.3 - Gráfico Mflops X Iterações (0 à 700)

4.5 COMPARAÇÃO ENTRE O CRAY E ESTAÇÕES SUN

No quarto e último teste realizou-se uma comparação entre o desempenho alcançado com o CRAY Y-MP e o desempenho alcançado com vários tipos de estações SUN. Nesta análise foi realizado o produto entre matrizes quadradas de 500 linhas e 500 colunas em diversas estações SUN, utilizando-se o programa apresentado na figura 4.4.

O programa mede o tempo necessário para realizar o produto matricial nas estações SUN e foi executado nos seguintes equipamentos: SUN 4/75, SUN 4/65, SUN 4/40 e SUN 4/20. A tabela 4.5 apresenta os resultados deste teste.

A primeira coluna da tabela 4.5 mostra os equipamentos utilizados na análise, a segunda coluna apresenta os tempos obtidos no produto matricial (matrizes quadradas de 500 linhas e 500 colunas) e a terceira coluna a proporção em relação ao CRAY.

Analisando-se os resultados na tabela 4.5 pode-se verificar uma significativa diferença de desempenho entre o CRAY e as estações SUN. No entanto, deve-se ressaltar que o programa utilizado no teste é adequado a vetorização, o que possibilita ao CRAY alcançar alto desempenho aproveitando as características especiais de sua arquitetura. Os valores apresentados na tabela 4.5 representam um problema específico, não podendo-se generalizar os resultados obtidos para todas as aplicações.

```

#include <sys/time.h>

main()
{
    struct timeval  tv;
    struct timezone tz;
    double          ti, tf, tr;
    float           tmp,
                  a[500][500],
                  b[500][500],
                  c[500][500];
    register int    i, j, k;

    gettimeofday(&tv,&tz);
    tmp = tv.tv_usec;
    ti = tv.tv_sec + (tmp / 1000000);
    for(i = 0;i < 500;i++)
        for(j = 0;j < 500;j++)
            {
                c[i][j] = 0;
                for(k = 0;k < 500;k++)
                    c[i][j] = c[i][j] + a[i][k] * b[k][j];
            }
    gettimeofday(&tv,&tz);
    tmp = tv.tv_usec;
    tf = tv.tv_sec + (tmp / 1000000);
    tr = tr - ti;
    printf("Tempo em segundos: %f\n",tr);
}

```

Figura 4.4 - Programa utilizado nos testes das estações SUN

TABELA 4.5 - COMPARAÇÃO ENTRE O CRAY E ESTAÇÕES SUN

EQUIPAMENTO	TEMPO(S)	PROPORÇÃO
CRAY Y-MP	1.16	1
SUN 4/75	271.81	234.32
SUN 4/65	476.94	411.16
SUN 4/40	485.83	418.84
SUN 4/20	656.70	566.12

5 CONCLUSÃO

Nos últimos anos os supercomputadores vêm assumindo uma posição de destaque junto a comunidade acadêmica e empresarial. Atualmente, o alto poder computacional disponibilizado através de suas características especiais de *hardware* e *software* é vital para diversas áreas de pesquisa e de grande importância para o desenvolvimento de diversos produtos.

Este trabalho discute várias características dos supercomputadores CRAY, as quais podem ser utilizadas para aperfeiçoamento dos programas, visando a obtenção do máximo desempenho. Os resultados obtidos nos testes demonstram que determinadas características organizacionais do programa influenciam de forma significativa no seu desempenho. Os principais fatores que influenciaram nos

resultados foram: conflitos de memória, número de iterações do laço mais interno (tamanho do vetor) e o preenchimento dos registradores vetoriais.

Durante os testes alcançou-se o índice máximo de 222.30 Mflops, o que apesar de representar um bom desempenho, manteve-se ainda distante do valor nominal de 330 Mflops de um processador do CRAY Y-MP. Deve-se ressaltar ainda, que a maioria dos resultados obtidos durante os testes conferem com os resultados apresentados na bibliografia sobre processamento paralelo e nos manuais da CRAY Research Inc.

Finalmente, torna-se importante salientar que os resultados apresentados neste trabalho representam o comportamento dos equipamentos quando submetidos a um problema específico, ou seja, o produto matricial. Os resultados não podem ser generalizados para todas as aplicações e devem ser considerados dentro do universo proposto pelo trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

- [EAR 92a] BARBOSA, Jorge Luis Victória; BARBOZA, Luciano Vitoria. **Métodos de Vetorização**. Pelotas, Curso de Pós-Graduação em Ciência da Computação na Universidade Católica de Pelotas, trabalho da disciplina *Tópicos Especiais em Matemática Aplicada e Computacional*, 1992.
- [EAR 92b] BARBOZA, Luciano Vitoria; BARBOSA, Jorge Luis Victória. **Algoritmos para o Produto Matricial em Máquinas SIMD**. Pelotas, Curso de Pós-Graduação em Ciência da Computação na Universidade Católica de Pelotas, trabalho da disciplina *Tópicos Especiais em Matemática Computacional*, 1992.
- [BAR 93] BARBOSA, Jorge Luis Victória. **Supercomputadores CRAY Y-MP**. Porto Alegre, Curso de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul, trabalho da disciplina *Arquitetura de Computadores para Processamento Paralelo*, 1993.
- [CRA 91a] **Cray Standard C Programmer's, Reference Manual**. SR-2074 3.0. Cray Research Inc, 1991.
- [CRA 91b] **UNICOS Performance Utilities, Reference Manual**. SR-2040 6.0. Cray Research Inc, 1991.
- [HWA 85] HWANG, Kai; BRIGGS, Fayé A. **Computer Architecture and Parallel Processing**. New York, McGraw-Hill Book Company, 1988.
- [NAV 90] NAVAUX, Philippe O. A. **Processadores Pipeline e Processamento Vetorial**. São Paulo, VII Escola de Computação, 1990.
- [QUI 87] QUINN, Michael J. **Designing Efficient Algorithms for Parallel Computers**. New York, McGraw-Hill Book Company, 1987.
- [ZIM 91] ZIMA, Hans; CHAPMAN, Barbara. **Supercompilers for Parallel and Vector Computers**. New York, Addison-Wesley Publishing Company, 1991.